Designing 3D RNA Origami Nanostructures with a Minimum Number of Kissing Loops

Antti Elonen
Department of Computer Science, Aalto University, Finland

Pekka Orponen
Department of Computer Science, Aalto University, Finland

Abstract

We present a general design technique for rendering any 3D wireframe model, that is any connected graph linearly embedded in 3D space, as an RNA origami nanostructure with a minimum number of kissing loops. The design algorithm, which applies some ideas and methods from topological graph theory, produces renderings that contain *at most one* kissing loop for many interesting model families, including for instance all fully triangulated wireframes. The design method is already implemented and available for use in the design tool DNAforge (`https://dnaforge.org`).

2012 ACM Subject Classification Theory of computation → Theory and algorithms for application domains; Applied computing → Life and medical sciences → Computational biology

Keywords and phrases RNA origami, wireframe nanostructures, polyhedra, kissing loops, topological graph embeddings, self-assembly

1 Background

Concurrently to the advances in DNA nanotechnology, there has been increasing interest in using RNA as the fabrication material for self-assembling bionanostructures. In comparison to DNA, the appeal of RNA is that the strands can be produced by the natural process of polymerase transcription, and the structures can thus be created in room temperature *in vitro*, and possibly eventually *in vivo*, from genetically engineered DNA templates. The challenge, on the other hand, is that the folding process of RNA is kinetically more complex and hence less predictable than DNA helix formation, at least at the present stage of RNA engineering.

Starting from the mid 1990's, the leading design technique in this area of *RNA nanotechnology* has been "RNA tectonics", whereby naturally occurring RNA structures are connected together with connector motifs such as kissing-loop and sticky-end pairings, to create complex target structures [11, 12]. A complementary top-down *de novo* design approach of "RNA origami" was however presented in a landmark 2014 article by Geary et al. [9]. In this method, broadly speaking, a given mesh model is rendered in RNA by designing a strand that will, firstly, fold upon itself to realise a spanning tree of the mesh by edges constituted as RNA helices, and secondly, induce the remaining edges by kissing-loop motifs that connect matching half-edge hairpin loops at 180° angles to create almost perfect "pseudo-helices". (This abstract view in terms of mesh models and spanning trees is from [16] and ignores many important details of the original work.) Following article [9], which demonstrated the feasibility of the RNA origami design method by the experimental synthesis and characterisation of several types of 2D RNA tiles, this line of work has been further developed with new connector motifs, design techniques, and tools in e.g. publications [15, 8].

Wireframe 3D RNA origami

The RNA origami idea has also been extended to cover 3D wireframe models [2]. (While article [2] addresses primarily polyhedral meshes, the method therein applies in fact to any connected straight-line wireframe model; that is, the meshes do not need to contain faces.) The basic spanning-tree based 3D design scheme is presented in Figure 1.

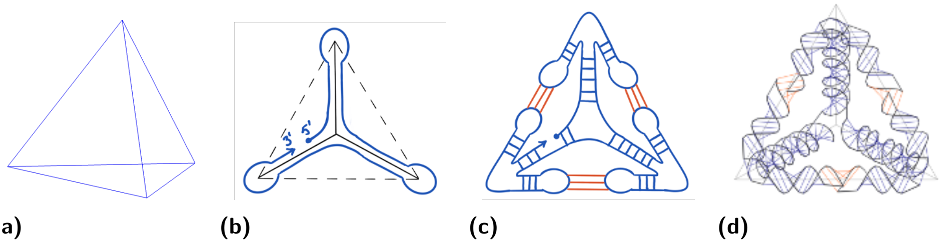


Figure 1 A spanning-tree based design scheme for 3D RNA wireframe origami. (a) Targeted wireframe model. (b) A spanning tree and strand routing of the wireframe graph. (c) Routing-based stem and kissing-loop pairings. (d) Helix-level model. (Adapted with permission from [2].)

In this scheme, one starts from the targeted wireframe, which in the case of Figure 1(a) is a simple tetrahedron. (Or more precisely the wireframe skeleton of a tetrahedral mesh.) In the first design step (Figure 1(b)) one chooses some spanning tree T of the wireframe graph G, and designs the primary structure of the RNA strand so that it folds to create a twice-around-the-tree walk on T, covering each edge of T twice in antiparallel directions. In the second design step (Figure 1(c)) one then extends the walk halfway along each of the co-tree (= non-spanning tree) edges of G into a hairpin loop, and designs the base sequences at the termini of the hairpins so that pairwise matching half-edges connect to form the 180° kissing-loop motifs mentioned earlier, thus constituting the co-tree edges. Figure 1(d) presents a helix-level model of the eventual nanostructure. (A similar design idea, although with different connector motifs and in the context of RNA-DNA hybrid nanostructures, has been recently applied also in the article [17].)

Challenges with kissing loops, goals of present work

Since the spanning tree of a connected graph with n vertices and m edges contains $n-1$ edges, its co-tree contains $m-n+1$ edges, and this is the number of kissing-loop connections employed by the previous method. While the method thus in principle applies to all connected 3D wireframe models, in practice using a large number of kissing-loop pairs in the designs raises some concerns. Firstly, kissing-loop pairings, which in the case of the 180° connector motif contain only six nucleotide pairs, may not be stable over long time scales. Secondly, the presence of a large number of slowly-forming tertiary structures such as kissing loops increases the risk of nonspecific pairings across structures, and hence aggregation of particles, in the synthesis stage. (There is some evidence of this in the experimental data presented in article [2].) And thirdly, there is at present no experimental data on large families of "orthogonal" kissing-loop pairs (high specific/low nonspecific pairing affinity) that would be needed for the design of complex structures using this method, and it is not even clear how large such families could reasonably be (cf. supplementary section S1.3.2. of article [2]).

Thus, in the present work we address the task of minimising the number of kissing loops in 3D RNA origami wireframe designs. As an application of an intimate connection between oriented strand routings on wireframes and topological graph embeddings, and building on earlier work from different contexts [21, 6, 4], we derive a polynomial time strand-routing algorithm that goes beyond the simple twice-around-the-tree idea, and minimises the number of kissing-loop connections needed to complete the design. As it turns out, the minimum number of kissing loops needed is *at most one* for many interesting classes of models, including for instance all fully triangulated wireframes. The method is already implemented and easily accessible in the online design tool DNAforge [3].

In the following, Section 2 presents the tight connection between viable strand routings and graph embeddings, and Section 3 the ensuing kissing-loop minimising strand routing algorithm. Section 4 introduces some graph classes where the maximum number of kissing loops is at most one, Section 5 discusses the DNAforge tool, and Section 6 provides a summary and some notes on further research directions.

2 Strong antiparallel traces and topological graph embeddings

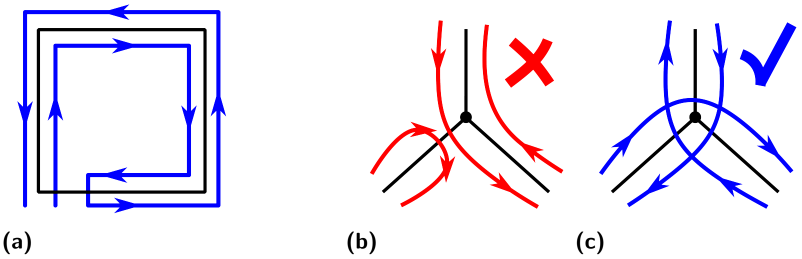


Figure 2 Strand routing criteria for RNA nanostructure design. (a) Edges covered twice in antiparallel directions. (b) Unstable vertex crossover pattern. (c) Stable vertex crossover pattern.

Let us first consider the possibility of rendering a given (connected) wireframe model using a single RNA strand with *no* kissing loops. This entails two conditions for the routing of the strand: firstly, every edge of the wireframe model must be covered twice, in antiparallel directions (Figure 2(a)); and secondly, the strand crossover pattern at each vertex must be *stable* (Figure 2(b)). The second condition signifies that if at a given vertex v with incident edges $e_1, \ldots, e_d$, one considers edges e_i and e_j to be *locally coupled* when there is a strand segment that crosses from e_i to e_j or vice versa, then this local edge coupling (multi-)graph must be connected; and since by the first condition it is regular of degree 2, it must be a cycle. (In the literature, the local routing pattern of the strands at a vertex is called a “transition” in [5, 1] and the local edge-connectivity graph the “vertex figure” in [4].)

Thus, every viable RNA strand routing of a wireframe model corresponds to an antiparallel double trace of its edges, in such a way that the edge-to-edge crossings at each vertex follow some local cyclic order, viz. a cyclic permutation of the incident edges. As it turns out, these conditions are exactly equivalent to the respective abstract graph (that is, the model with geometry ignored) having *a 1-face cellular embedding in some orientable surface*, a result established by Fijavž et al. in 2014, albeit in the context of polypeptide nanostructure designs [4]. Fijavž et al. call graph walks that satisfy the two indicated conditions *strong antiparallel traces*. (Earlier studies along the same lines, but not quite establishing the same connection, include e.g. [20, 19, 1, 13].)

Unfortunately, graphs that contain strong antiparallel traces are not that common, as observed already with an incomplete characterisation in [13]. Notably e.g. all of the Platonic solids are counterexamples, and thus cannot be properly rendered with a single RNA strand.

As we shall see, however, admitting even a single kissing loop in the designs changes the situation dramatically.

Surfaces, graph embeddings, and Euler's formula

To get a proper understanding of the methodology, let us review some key topology concepts and results about surfaces and graph embeddings [14].

- A *surface* S is a topological space of dimension two (a 2-manifold), meaning that every point in the space has a neighbourhood homeomorphic to an open unit disk. (A *homeomorphism* is a topological isomorphism, precisely speaking a continuous bijection between two topological spaces with a continuous inverse.)
- A surface S is *orientable* if there is a consistent sense of clockwise/counterclockwise at each point of S; technically speaking if there is no embedding of the Möbius strip in S.
  - We shall only be considering surfaces that are connected, orientable, topologically compact and without boundary. This class of surfaces includes e.g. the sphere and the torus, but not e.g. either the open disk (not compact) or the closed disk (has boundary), and of course not nonorientable surfaces such as the Möbius strip or the Klein bottle.
  - *From now on, the word "surface" in this paper means a connected, orientable, compact surface without boundary, unless otherwise explicitly stated.*
- The *genus* of a surface S is the number of nonintersecting cycles that can be drawn on S without separating it.

The classification theorem of surfaces states that every (connected, orientable, compact, boundaryless) surface is homeomorphic to either the sphere or k torii sewn together (intuitively the surface of a "k-hole donut"), for some $k \geq 1$. Furthermore, since the sphere has genus 0 and a k-torus has genus k, for this family of surfaces the genus is a topological invariant: any two surfaces with the same genus are homeomorphic, and vice versa.

- An *embedding* of a graph $G = (V, E)$ in a surface S is a continuous 1-1 mapping of G into S as a system of points and arcs. (That is, the vertices V get mapped into points in S, and the edges E into corresponding point-connecting arcs, in such a way that the arcs don't cross in S.)
- An embedding $\epsilon : G \to S$ divides S (or, technically, $S \setminus \epsilon(G)$) in disjoint *regions* or *faces*. If every face is homeomorphic to an open disk, the regions are called *cells* and the embedding is a *cellular embedding*.
- Any cellular embedding of a graph $G = (V, E)$ in a surface of genus g satisfies *Euler's generalised polyhedral formula*, or briefly just *Euler's formula*:

  $$|V| - |E| + |F| = 2 - 2g,$$

  where $|F|$ is the number of cellular faces in the embedding.

To illustrate these concepts, let us consider the simple example of embedding the cube graph. Figure 3(a) presents a "natural" cellular embedding of this graph in a sphere surface. The embedding comprises six cells that correspond to the six faces of the 3D cubical polyhedron. Figure 3(b) illustrates how the corresponding cubical wireframe could be assembled using six RNA or DNA strands, each routing one of the faces of the cube cyclically in a counterclockwise direction. (The strand that routes the front face is indicated separately by a dotted line.) Figure 3(c) presents the same strand routing projected on the planar

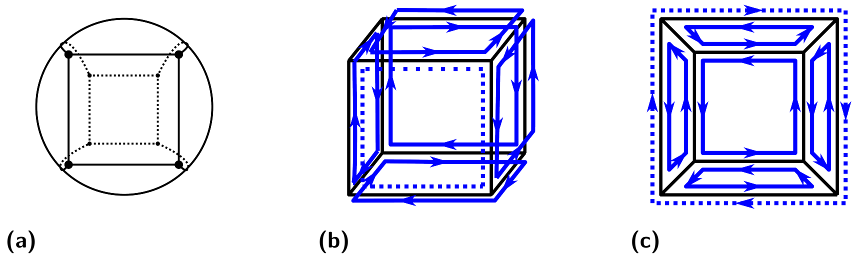


Figure 3 Cube graph embedded in a sphere surface. (a) Visualisation of the embedding. (b) Counterclockwise strand cycles routing the faces of the cube polyhedron model; each edge covered twice in antiparallel directions. (c) Faces and cycle routings presented in the Schlegel diagram of the model.

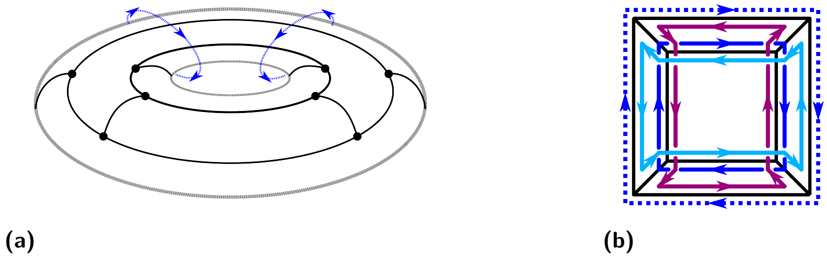


Figure 4 Cube graph embedded in a torus surface. (a) Visualisation of the embedding. (b) Corresponding strand routings presented in the Schlegel diagram, with each edge covered twice in antiparallel directions.

Schlegel diagram of the polyhedron. Since the sphere has genus $g = 0$, one can validate that Euler's formula holds: $|V| - |E| + |F| = 8 - 12 + 6 = 2 = 2 - 2g$.

However, the cube graph can also be cellularly embedded in a torus surface as presented in Figure 4(a). Now there are only four cells, and a corresponding system of four strands cyclically routing the cells, again in counterclockwise orientation and covering each of the graph edges twice in antiparallel directions, is outlined in the Schlegel diagram in Figure 4(b). For added clarity, Figure 5(a) indicates the four cells labelled as A, B, C, D, and Figure 5(b) shows the cell partioning and the strand routings on a 2D torus diagram, which “folds around” at the top/bottom and left/right boundaries. Again Euler's formula can be validated, now with the toroidal genus $g = 1$: $|V| - |E| + |F| = 8 - 12 + 4 = 0 = 2 - 2g$.

Note that in both of these cube graph embeddings, the strand crossovers at the vertices follow some cyclic order; in both cases actually the clockwise order around each vertex in the

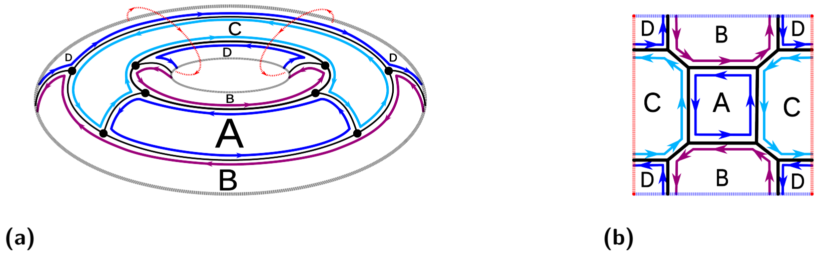


Figure 5 Cube graph embedded in a torus surface. (a) Labelled visualisation of the embedding. (b) Cell partitions and strand routings displayed on a torus diagram.

respective embedding surface. Conveniently, this arrangement (i) ensures that the vertices are *stable* in the sense defined earlier, and (ii) results in counterclockwise routes around the cells, which also guarantees that the cell boundaries, viz. the graph edges, are all covered twice in antiparallel directions.

In fact, any cellular embedding of a graph G in an orientable surface S induces such a system of (relative clockwise) permutations of incident edges at each vertex of G, that uniquely determines the embedding. And vice versa: any system of local edge permutations at the vertices of a graph G that also guarantees antiparallel coverage of the edges corresponds to some embedding in an orientable surface.

Note also that the number N of cyclic strands required to fabricate a graph $G = (V, E)$, or the corresponding metric wireframe, according to the recipe provided by a given cellular embedding equals the number of faces $|F|$ in that embedding. Thus, by Euler's formula, this number and the genus of the embedding surface are in an inverse relationship:

$$N = |F| = |E| - |V| + 2 - 2g.$$

Thus, to minimise the number of strands needed, one should find an embedding into a surface of maximum possible genus. Ideally, one would hope to achieve $N = 1$, that is a cellular embedding in a surface of genus $g_{\text{ideal}} = \frac{1}{2}(|E| - |V| + 1)$ that comprises *a single face*. The cyclic strand route around this face would then constitute a strong antiparallel trace of the graph G.

3 The Xuong tree design method

As discussed earlier, many interesting graphs do not admit strong antiparallel traces, or equivalently single-face cellular embeddings of the absolutely maximum genus g_{max}. However, in the context of RNA origami design one can compromise on this target by judiciously removing some edges from the target graph G so as to reach the maximum *achievable* single-face embedding genus, and then reintroducing the removed edges as kissing loops. This is the idea underlying our *Xuong tree design method* for RNA origami, to be presented next.

Let $G = (V, E)$ be a connected graph, T a spanning tree of G, and $co(G, T) = G \setminus T$ the co-tree of G corresponding to T. All the spanning trees of G are of size (= number of edges) $|V| - 1$ and all the co-trees correspondingly of size $|E| - |V| + 1$. The latter value is called the *Betti number*, or cycle rank, of G.

The *Betti deficiency* $\xi(G, T)$ of a spanning tree T in G is defined to be the number of odd-sized components of $co(G, T)$. The *deficiency of a graph* G is the minimum Betti deficiency over all its spanning trees, $\xi(G) = \min_T \xi(G, T)$ [6, 21].

▶ **Theorem 1** (Xuong 1979 [21]). *The maximum achievable embedding genus of a graph G is $\gamma(G) = \frac{1}{2}(\beta(G) - \xi(G))$.*

[Note that, in reference to the previous section, $\gamma(G) = g_{\text{ideal}} - \frac{1}{2}\xi(G)$, hence the term "deficiency" $\xi(G)$.]

A spanning tree T^* that realises Theorem 1, that is for which $\xi(G, T^*) = \xi(G)$, is called a *Xuong tree*, and the maximum genus embedding of G can be found by constructing it around T^* in a process described in [21]. A fundamental operation in this process is inserting graph edges into faces, as represented by their boundary walks (= cycles in which edges may be repeated) in adjacent pairs. Consider two vertices, v_1 and v_2. If v_1 and v_2 are along the boundary walk of a single face, adding a new edge between them results in the face, as represented by the walk, splitting in two (Figure 6b). Conversely, if v_2 and v_3 are not

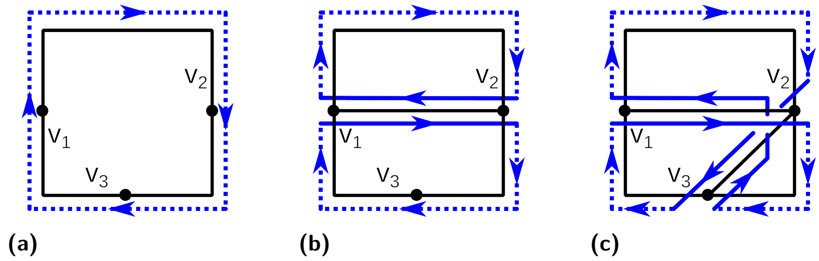


Figure 6 Splitting and joining faces by inserting edges. **(a)** A single face and its boundary walk. **(b)** The face/boundary walk is split in two by the insertion of an edge in the walk between vertices v_1 and v_2. **(c)** The two faces/walks are merged into one with the insertion of another edge between v_2 and v_3.

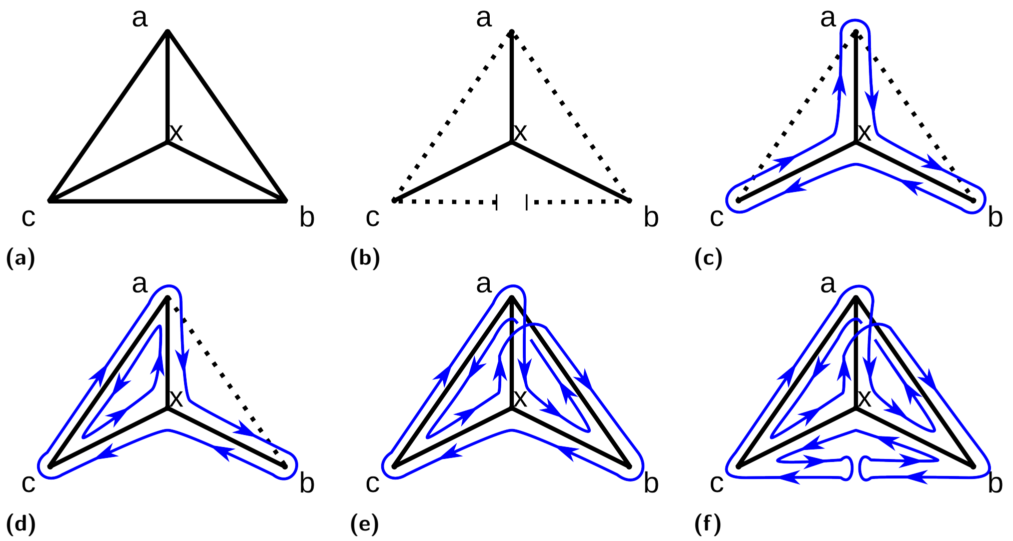


Figure 7 The Xuong-tree RNA wireframe design method applied on a tetrahedron. **(a)** A Schlegel diagram of a tetrahedron. **(b)** The tetrahedron with one edge removed and marked as a kissing loop. The solid line represents a Xuong tree. **(c)** The initial route along the Xuong tree: $a \rightarrow x \rightarrow b \rightarrow x \rightarrow c \rightarrow x \rightarrow a$ **(d)** The partial route augmented with a new edge $\{c, a\}$ from the co-tree. Note that the cycle count is now increased to two: $a \rightarrow x \rightarrow b \rightarrow x \rightarrow c \rightarrow a$ and $a \rightarrow c \rightarrow x \rightarrow a$. **(e)** After inserting the adjacent edge $\{a, b\}$, cycle count drops back to one: $a \rightarrow x \rightarrow b \rightarrow a \rightarrow c \rightarrow x \rightarrow a \rightarrow b \rightarrow x \rightarrow c \rightarrow a$ **(f)** The final tetrahedron with the deleted edge reintroduced as a kissing loop.

along the boundary walk of the same face, inserting an edge connecting them merges the two walks, i.e. faces into one (Figure 6c).

Since the boundary walk around a tree constitutes a single face, one can start with a 1-face embedding of T^* and then add two mutually adjacent edges from $co(T^*)$ to T^* and preserve the 1-face embedding. This process can be continued until all the even-size components in $co(T^*)$ are exhausted, and since T^* is specifically chosen to be a spanning tree of G such that $co(T^*)$ has the minimum number of odd components, a maximum genus embedding is found.

A Xuong tree T^* serving as a starting point for this process can be found in polynomial time by a reduction to the matroid parity problem for which polynomial time algorithms exist [18, 7]. This approach for finding Xuong trees was presented by Furst et al. [6], who also provide a time complexity bound of $O(mnd\log^6 n)$ for the method, where m is the number of edges in the graph G, n is the number of vertices, and d the maximum vertex degree.

In case the graph G does not have a single-face embedding, it can be modified to have one by removing k edges, one edge for each odd component of $co(T^*)$. It is not possible to transform it into a single-face embeddable graph by removing any fewer edges than k. Suppose that removing $k-1$ edges from G resulted in a single-face embeddable graph G_t. In that case, $\gamma(G_t) = \frac{2-|V|+|E_t|-|F_t|}{2} = \frac{2-|V|+|E|-k}{2} > \gamma(G) = \frac{2-|V|+|E|-|F|}{2} = \frac{2-|V|+|E|-k-1}{2}$, which is a contradiction, since G_t, a subgraph of G, cannot have a higher genus embedding than the highest genus embedding of G itself.

The Xuong tree design method routes the RNA strand around the single face embedding of the modified graph and replaces the removed edges with kissing loops. The number of kissing loops required is also minimised by the same argument as before. The Xuong tree design method is illustrated in figure 7 for a tetrahedron. Since a tetrahedron has 4 vertices and 6 edges, both all its spanning trees and co-trees have 3 edges, which means that any co-tree has at least one odd-sized component and a tetrahedron is not single face embeddable. (In fact all the co-trees of a tetrahedron are connected and of size 3.) By removing one edge, however, the tetrahedron can be 1-face embedded. The Xuong tree design method will then find a Xuong tree of the modified tetrahedron and use it to construct 1-face embedding. The removed edge is reintroduced as a kissing loop in the final step.

4 Upper embeddable graphs

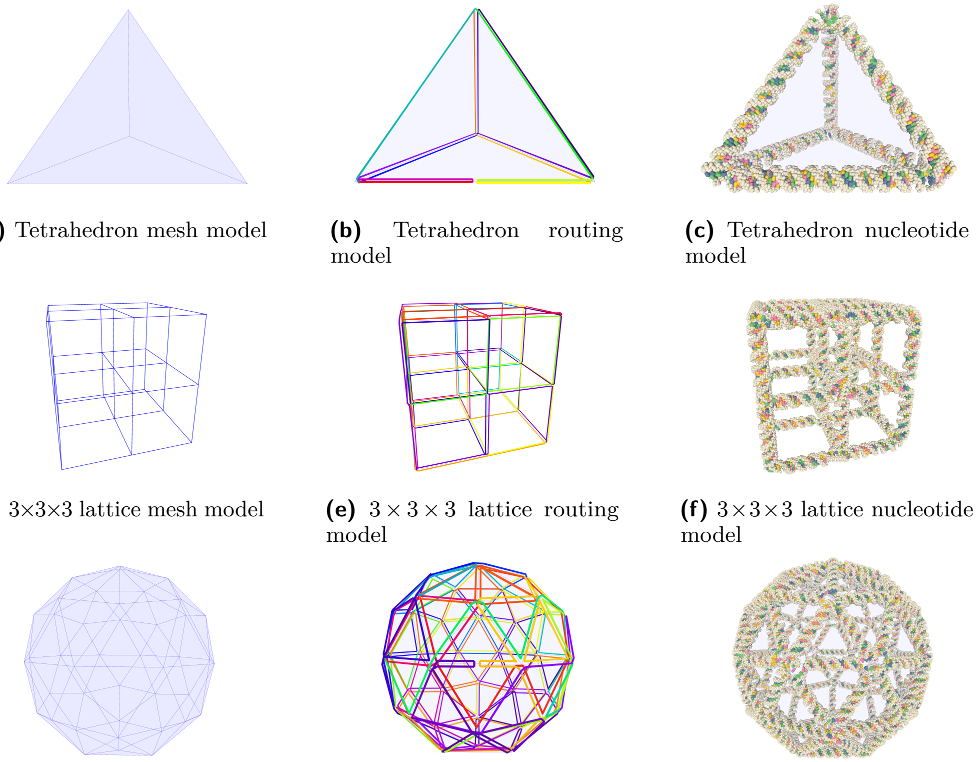
Graphs that readily admit a 1-face or a 2-face embedding are called *upper embeddable*. Such graphs require at most only one kissing loop using the Xuong tree design method. Many interesting graph classes are upper embeddable, including the following ones listed by Gross et al. [10, p. 752]

- Locally connected graphs,
- Cyclically edge-4-connected graphs,
- k-regular vertex-transitive graphs of girth g with $k \geq 4$ or $g \geq 4$,
- Loopless graphs of diameter 2,
- $(4k+2)$-regular graphs and $(2k)$-regular bipartite graphs.

A graph G is *locally connected*, if for every vertex v in G, the open neighbourhood of v (vertices adjacent to v, excluding v) induces a connected graph in G. This graph class being upper embeddable is of particular interest, since it covers all fully triangulated polyhedra, e.g., tetrahedron, octahedron, icosahedron, and countless more. A graph G is said to be *vertex transitive*, if for every pair of vertices in G, there exists an automorphism mapping one vertex to the other. Intuitively, this means that the graph looks the same from the point of view of any individual vertex. All Platonic solids have this property, and, since cube and dodecahedron both have a girth (shortest cycle length) ≥ 4, they too are upper embeddable.

5 The DNAforge design tool

DNAforge [3] is an online platform for designing DNA and RNA wireframe nanostructures from 3D models. The Xuong tree design method is integrated in DNAforge as XT-RNA. It allows users to transform any connected wireframe 3D model into a nucleic acid nanostructure with a single click. The workflow for the XT-RNA method is depicted in figure 8 for a tetrahedron, a $3 \times 3 \times 3$ cubical lattice, and a sphere. Note that the tetrahedron and the



(a) Tetrahedron mesh model

(b) Tetrahedron routing model

(c) Tetrahedron nucleotide model

(d) 3×3×3 lattice mesh model

(e) $3 \times 3 \times 3$ lattice routing model

(f) 3×3×3 lattice nucleotide model

(g) Sphere mesh model

(h) Sphere routing model

(i) Sphere nucleotide model

Figure 8 The XT-RNA design workflow in *DNAforge* for a tetrahedron (**(a)**, **(b)**, and **(c)**), a 3×3×3 lattice (**(d)**, **(e)**, and **(f)**), and a sphere (**(g)**, **(h)**, and **(i)**).

sphere require only one kissing loop, presented in the foreground, whereas the lattice does not require any.

The DNAforge interface gives users options to minimise strain in the designed nanostructure via duplex-level physical simulation, or to simulate it directly from the interface with a nucleotide level simulation, provided the DNAforge backend is installed. The primary sequence is currently generated randomly, subject to Watson-Crick pairing conditions, and the routing is also currently found using randomised spanning trees instead of the matroid-reduction algorithm. In our experience, the randomised search works well, suggesting that typical graphs have many Xuong trees.

The final design can be exported as a PDB file, UNF file, or as OxDNA files, and the primary sequence can be exported as a CSV file.

6 Conclusion

Discussion and further work

We have introduced a general single-stranded RNA routing method, which minimises the use of kissing loops, and is based on high genus graph embeddings. The XT-RNA implementation is available at `https://dnaforge.org/`.

Note that even though the Xuong tree designs typically have at most only one kissing loop, the base pairing structure along the helices constituting the wireframe edges is highly pseudoknotted. Development opportunities for the Xuong tree design method include addressing the challenges posed by this pseudoknotted nature of its designs. These challenges are related to the primary sequence generation and the knottedness of the actual nucleic acid strands. While a strong antiparallel double trace of a wireframe model embedded on a sphere-equivalent surface can be guaranteed to be an unknot, there are no such guarantees for higher-genus wireframe models, and taking the helicity of RNA duplexes into account will introduce knots even for the lower-genus models. These problems could potentially be addressed by optimising the number and placement of kissing loops. Since RNA strands are not closed cycles, however, the problem of knottedness might not be detrimental, and practical experimentation is warranted.

References

1 Joanna A Ellis-Monaghan. Transition polynomials, double covers, and biomolecular computing. *Congressus Numerantium*, 166:181, 2004.

2 Antti Elonen, Ashwin Karthick Natarajan, Ibuki Kawamata, Lukas Oesinghaus, Abdulmelik Mohammed, Jani Seitsonen, Yuki Suzuki, Friedrich C. Simmel, Anton Kuzyk, and Pekka Orponen. Algorithmic design of 3D wireframe RNA polyhedra. *ACS Nano*, 16:16608–18816, 2022. doi:10.1021/acsnano.2c06035.

3 Antti Elonen, Leon Wimbes, Abdulmelik Mohammed, and Pekka Orponen. DNAforge: A design tool for nucleic acid wireframe nanostructures. *Nucleic Acids Research*, to appear. preprint: https://research.cs.aalto.fi/nc/papers/dnaforge_2024.pdf. doi:10.1093/nar/gkae367.

4 Gašper Fijavž, Tomaž Pisanski, and Jernej Rus. Strong traces model of self-assembly polypeptide structures. *MATCH Communications in Mathematical and in Computer Chemistry*, 71:199–212, 2014. preprint: https://doi.org/10.48550/arXiv.1308.4024.

5 Herbert Fleischner. *Eulerian Graphs and Related Topics. Part 1, Volume 1*, volume 45 of *Annals of Discrete Mathematics*. North-Holland Publishing Co., Amsterdam, 1990.

6 Merrick L. Furst, Jonathan L. Gross, and Lyle A. McGeoch. Finding a maximum-genus graph imbedding. *Journal of the ACM (JACM)*, 35(3):523–534, 1988. doi: 10.1145/44483.44485.

7 Harold N Gabow and Matthias Stallmann. Efficient algorithms for graphic matroid intersection and parity. In *Automata, Languages and Programming: 12th Colloquium Nafplion, Greece, July 15–19, 1985 12*, pages 210–220. Springer, 1985. doi:10.1007/BFb0015746.

8 Cody Geary, Guido Grossi, Ewan K. S. McRae, Paul W. K. Rothemund, and Ebbe S. Andersen. RNA origami design tools enable cotranscriptional folding of kilobase-sized nanoscaffolds. *Nature Chemistry*, 13(6):549–558, 2021. doi:10.1038/s41557-021-00679-1.

9 Cody Geary, Paul W. K. Rothemund, and Ebbe S. Andersen. A single-stranded architecture for cotranscriptional folding of RNA nanostructures. *Science*, 345(6198):799, 2014. doi:10.1126/science.1253920. URL: https://science.sciencemag.org/content/345/6198/799.abstract, doi:10.1126/science.1253920.

10 Jonathan L. Gross, Jay Yellen, and Ping Zhang. *Handbook of Graph Theory*. CRC Press, 2nd edition, 2013.

11 Peixuan Guo. The emerging field of RNA nanotechnology. *Nature Nanotechnology*, 5(12):833–842, December 2010. doi: 10.1038/nnano.2010.231.

12 Daniel Jasinski, Farzin Haque, Daniel W. Binzel, and Peixuan Guo. Advancement of the emerging field of RNA nanotechnology. *ACS Nano*, 11(2):1142–1164, 2017. doi: 0.1021/acsnano.6b05737.

13 Sandi Klavžar and Jernej Rus. Stable traces as a model for self-assembly of polypeptide nanoscale polyhedrons. *MATCH Communications in Mathematical and in Computer Chemistry*, 70(1):317–330, 2013. url: http://match.pmf.kg.ac.rs/content70n1.htm.

14 John Lee. *Introduction to Topological Manifolds*, volume 202. Springer Science & Business Media, 2010. doi:10.1007/978-1-4419-7940-7.

15 Di Liu, Cody W. Geary, Gang Chen, Yaming Shao, Mo Li, Chengde Mao, Ebbe S. Andersen, Joseph A. Piccirilli, Paul W. K. Rothemund, and Yossi Weizmann. Branched kissing loops for the construction of diverse RNA homooligomeric nanostructures. *Nature Chemistry*, 12(3):249–259, 2020. doi:10.1038/s41557-019-0406-7.

16 Abdulmelik Mohammed, Pekka Orponen, and Sachith Pai. Algorithmic design of cotranscriptionally folding 2D RNA origami structures. In *Unconventional Computation and Natural Computation: 17th International Conference, UCNC 2018, Fontainebleau, France, June 25-29, 2018, Proceedings 17*, pages 159–172. Springer, 2018. doi: 10.1007/978-3-319-92435-9_12.

17 Molly F. Parsons, Matthew F. Allan, Shanshan Li, Tyson R. Shepherd, Sakul Ratanalert, Kaiming Zhang, Krista M. Pullen, Wah Chiu, Silvi Rouskin, and Mark Bathe. 3D RNA-scaffolded wireframe origami. *Nature Communications*, 14(1):382, 2023. doi:10.1038/s41467-023-36156-1.

18 Matthias Stallmann and Harold N Gabow. An augmenting path algorithm for the parity problem on linear matroids. In *25th Annual Symposium on Foundations of Computer Science, 1984*, pages 217–228. IEEE, 1984. doi:10.1109/SFCS.1984.715918.

19 Carsten Thomassen. Bidirectional retracting-free double tracings and upper embeddability of graphs. *Journal of Combinatorial Theory, Series B*, 50(2):198–207, 1990. doi:10.1016/0095-8956(90)90074-A.

20 Martin Škoviera and Roman Nedela. The maximum genus of a graph and doubly eulerian trails. *Bollettino dell'Unione Matematica Italiana B*, 4 B:541–551, 1990.

21 Nguyen Huy Xuong. How to determine the maximum genus of a graph. *Journal of Combinatorial Theory, Series B*, 26(2):217–225, 1979. doi:10.1016/0095-8956(79)90058-3.